

# 電池残量にもとづく無線の中継機能の起動と停止による IoT デバイスの省電力化

杉本 一彦<sup>1</sup> 串田 高幸<sup>1</sup>

**概要：**IoT の発展からセンサーデータの多様化により屋外で利用するケースが増加している。屋外の利用では IoT デバイスに電池を搭載しメッシュネットワークを用いる手法がある。しかしセンサーデータを転送する必要のない場合に中継機能を起動していることによって削減することができる消費電力を削減できない。この課題を解決するため中継機能を電池残量に基づいて起動と停止を行い、IoT デバイスの省電力化の手法を提案した。中継機能の起動と停止には電池残量がより多い IoT デバイスの中継機能を起動し、そうでない IoT デバイスの中継機能は停止する。評価では IoT デバイスの動作時間と消費電力を既存手法と比較した。既存手法と比較し動作時間は最大 180 秒増加し、消費電力は最大 16.2mAh 減少することが実現できた。

## 1. はじめに

### 背景

電池で動作する IoT デバイスにおいて、メッシュネットワークを構築する際、それぞれの IoT デバイスで消費電力が異なる場合がある。これはデータの転送量が異なるためである。メッシュネットワークを用いない場合、IoT デバイスは自らのセンサーデータのみ送信すれば良い。しかし、メッシュネットワークにてセンサーデータを送信する場合、別の IoT デバイスを経由してアクセスポイントへデータが到着する。そのためそのアクセスポイントとセンサーデータを送信する IoT デバイスの間に存在し、センサーデータの転送を行う IoT デバイスは消費電力が増加する [1]。

### 課題

電池で動作する IoT デバイスにてメッシュネットワークを構築する際に、センサーデータを転送する必要のない場合に中継機能を起動していることによって削減することができる消費電力を削減できないことが課題である。図 1 にて課題の例を示す。

図 1 では IoT デバイスが 3 台 (A, B, C) と Wi-Fi ルータ、また IoT デバイスのデータ送信先であるサーバがある。IoT デバイスは全て電池で動作している。また Wi-Fi ルータは外部電源に接続されている。IoT デバイスはサーバ

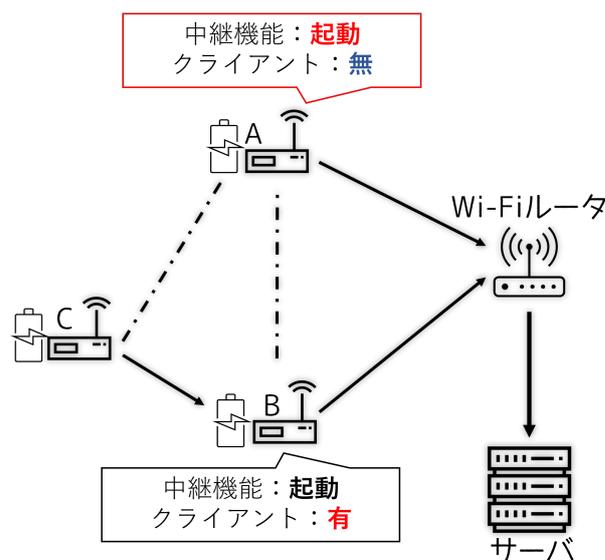


図 1 中継機能の有効化における消費電力の増加

へセンサーデータを送信することが目的で動作している。IoT デバイス C は Wi-Fi ルータへ直接接続することが出来ないため IoT デバイス A もしくは B を経由してデータを送信する必要がある。図 1 においては IoT デバイス C が IoT デバイス B に接続されている。この場合 IoT デバイス A のスタンバイ Wi-Fi はクライアントがない状態となる。しかし従来の手法ではクライアントがない場合であっても中継機能を有効にしている必要がある。従って中継機能を起動する必要のない状態で起動し続けているため、削減できるはずの消費電力が削減できない。

<sup>1</sup> 東京工科大学大学院バイオ・情報メディア研究科コンピュータサイエンス専攻  
〒192-0982 東京都八王子市片倉町 1404-1

## 2. 提案方式

本章では本研究での提案方式とそれを用いたユースケースシナリオについて述べる。

本稿における提案手法は中継機能を電池残量に基づいて起動と停止を行い, IoT デバイスの省電力化を行う。提案手法における前提条件は IoT デバイスは既に設置されているものとし, 追加と撤去を行わないものとする。またネットワークにおける最大のホップ数は 2 とする。

### 2.1 基礎実験

本節では課題で先述した中継機能の起動により消費電流が高くなることを確認するために基礎実験を実施した結果を示す。また通信時における消費電流や, 転送データ量における消費電流の比較を行う。

基礎実験で使用した IoT デバイスは ESP-WROOM-32 である。消費電流の測定は 0.5 秒に 1 回とし, 測定したセンサーは INA219 である。

図 2 は ESP-WROOM-32 における中継機能停止時 Wi-Fi への接続をしない場合, Wi-Fi へ接続している場合, 中継機能を起動している場合のそれぞれにおいて消費電流を測定した結果を示している。

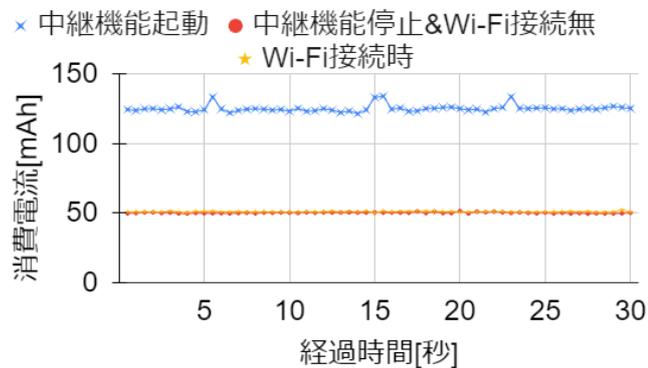


図 2 中継機能停止 Wi-Fi 接続, Wi-Fi 接続, 中継機能起動の消費電流の比較

図 2 における横軸は実験の経過時間, 縦軸は消費電流である。実験時間は 30 秒間である。データは 3 種類あり, 中継機能を停止し, かつ Wi-Fi への接続をしない (中継機能停止 Wi-Fi 接続無) 場合, 中継機能を起動している (中継機能起動) 場合, Wi-Fi へ接続しデータの送受信は行っていない場合 (Wi-Fi 接続時) である。中継機能を停止し, かつ Wi-Fi への接続をしない (中継機能停止 Wi-Fi 接続無) 場合における消費電流は最低 49.7mAh であり最高 51.6mAh である。中継機能を起動している (中継機能起動) 場合における消費電流は最低 50.1mAh であり最高 52.5mAh である。Wi-Fi へ接続しデータの送受信は行っていない (Wi-Fi 接続時) 場合における消費電流は最低 121.6mAh であり最高 134.1mAh である。図 2 より中継機能を停止, Wi-Fi へ

接続していない場合に比べ中継機能を起動している際の消費電流が最低 70mA, 最高 84mA 増加する。

図 3 はソケット通信でのデータ送受信時における消費電流を比較した結果を示している。ソケット通信は TCP にて行う。

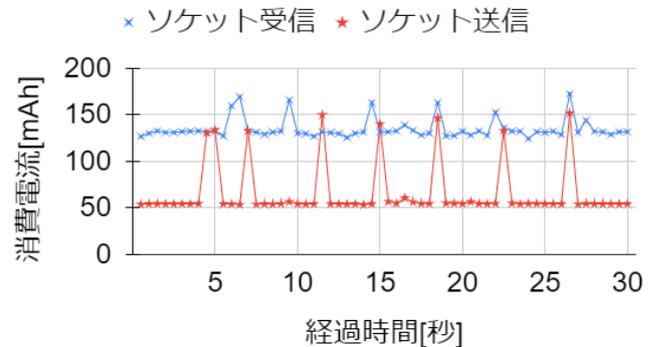


図 3 ソケット通信でのデータ送受信時における消費電流の比較

図 3 における横軸は実験の経過時間, 縦軸は消費電流である。実験時間は 30 秒間である。また図 2 と同じく消費電流は 0.5 秒に 1 回計測を行っており, 測定したセンサーは INA219 である。ソケット通信にてデータを 4 秒ごとにデータの送受信を行なっている。送受信におけるデータは ESP-WROOM-32 の固有の ID である 11Byte と温度データの 7Byte を足し合わせた合計 18Byte である。ソケット通信でのデータ受信時は中継機能を起動している。従ってソケット通信によるデータ受信時はデータ送信時と比較し消費電流が増加している。またデータ受信時とデータちらにおいても消費電流が一時的に増加している。しかし図 3 ではソケット通信によるデータ送受信のタイミングが正確な 4 秒間隔となっていない。これは実験で仕様した ESP-WROOM-32 の sleep 時間の精度が低いことが原因であると考えられる。

図 4 は図 2 と図 3 で計測した消費電流の平均値をグラフとして示したものである。

図 4 における横軸は ESP-WROOM-32 の動作状況, 縦軸は消費電流である。消費電流の平均は高い順に A. ソケット受信が 134.9mAh, B. 中継機能起動が 125.2mAh, C. ソケット送信が 66.0mAh, D. Wi-Fi 接続が 51.1mAh, E. 中継機能停止 Wi-Fi 接続無が 50.2mAh である。

次に転送するデータ量を 5 段階に分けて消費電力を測定した。1 つの転送データは合計 18 文字の文字列データである。データは ESP-WROOM-32 の固有の ID である 11Byte と温度データの 7Byte を足し合わせた合計 18Byte である。基礎実験では転送データ量を 0, 18, 36, 54, 72, 90Byte と変化させてそれぞれ消費電流を測定した。また転送データ量はペイロードのデータ量を指している。結果を図 5 に示す。

図 5 における横軸は転送データ量であり縦軸は消費電流

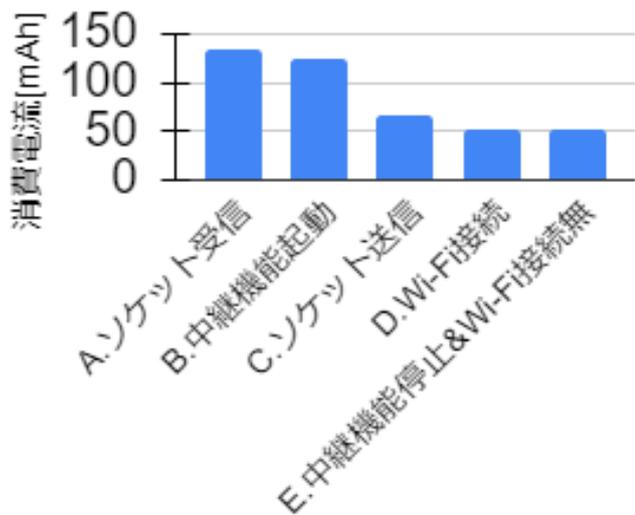


図 4 基礎実験における消費電流の平均値の比較

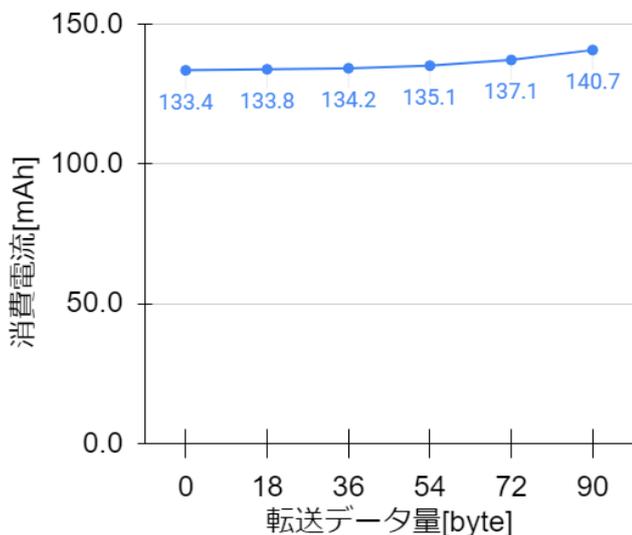


図 5 転送データ量と消費電流の関係

である。それぞれの転送データ量の消費電流は転送データ量が 0Byte である場合は 133.4mAh であり、18Byte の場合は 133.8mAh、36Byte の場合は 134.2mAh、54Byte の場合は 135.1mAh、72Byte の場合は 137.1mAh、90Byte の場合は 140.7mAh である。この結果より転送データ量が増加するにつれて消費電流は増加する傾向がある。しかし転送データ量が 0Byte と 90Byte の消費電流を比較した場合でも 7.2mAh の差である。

基礎実験の結果より転送データ量の増加より、中継機能が起動することの方が消費電流が増加する要因である。

### 提案方式

図 6 にて提案手法を説明する。

図 6 では IoT デバイス 5 台 (A, B, C, D, E) と Wi-Fi ルータ 1 台、サーバ 1 台で構築されているネットワークを示している。IoT デバイス C, D, E のデータを IoT デバイ

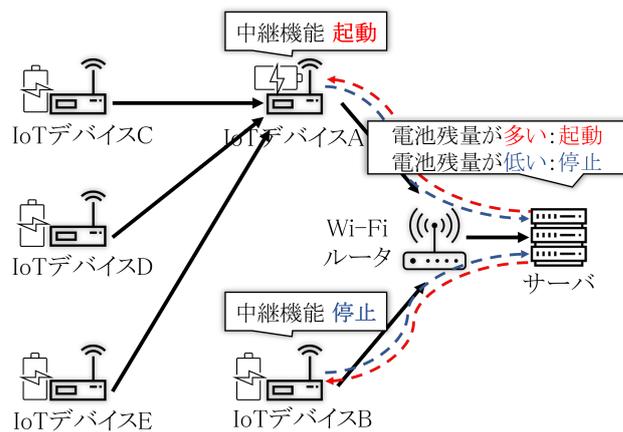


図 6 提案のイメージ図

ス A もしくは IoT デバイス B が転送する必要がある。転送する際は中継機能を起動する必要がある。またクライアントとなる IoT デバイス C, D, E は IoT デバイス A, B のどちらにも接続できる状況を想定している。

提案手法ではデータの転送を担う IoT デバイスの中継機能の起動と停止を行うことで消費電力の削減を行う。中継機能の起動と停止は電池残量を比較し、電池残量が多い IoT デバイスの中継機能を起動させ、より少ない IoT デバイスの中継機能を停止する。条件式を式 (1) に示す。

$$\begin{cases} P_a & (P_a - P_b > 0) \\ P_a & (P_a - P_b = 0) \\ P_b & (P_a - P_b < 0) \end{cases} \quad (1)$$

$P_a$  が IoT デバイス A の電池残量であり、 $P_b$  が IoT デバイス B の電池残量である。電池残量を比較し、 $P_a$  が  $P_b$  より多い場合は中継機能を起動する IoT デバイスは IoT デバイス A となる。 $P_b$  が  $P_a$  より多い場合は中継機能を起動する IoT デバイスは IoT デバイス B となる。また  $P_a$  が  $P_b$  が同じ値であった場合はどちらの IoT デバイスの中継機能を起動しても問題ない。しかし本提案においてはこの場合、中継機能を起動する IoT デバイスは IoT デバイス A とする。

図 6 においてはサーバにて電池残量の比較を行なっている。本手法によりクライアントとなる IoT デバイス C, D, E は中継機能が起動している一方の IoT デバイスへと接続しデータの送信を行う。

中継機能を停止することにより IoT デバイスの省電力化を実現する。また中継機能が起動している IoT デバイスは消費電力が高くなる。その結果データの転送を行う IoT デバイスの電池残量の順位が変わることがあるため、電池残量の順位が変わったタイミングで中継機能の起動と停止を切り替える。従ってどちらか一方の IoT デバイスの電池残量が低下していくことを防ぐ。

## ユースケース・シナリオ

本節では前節で先述した提案を用いたユースケースシナリオについて説明する。ユースケースシナリオでは水田における IoT デバイスを用いた水質監視システムの構築を想定する。水質監視は水量の測位と水温の計測を行う。本提案では電池にて動作する IoT デバイスの利用を想定している。そのため屋外において、かつ電力系統からの電源供給がなされず、インターネットへ接続するためのアクセスポイント (AP) を設置する、ネットワークインフラの構築が出来ない状況を考えている。以上の理由により水田での利用を策定した。

本ユースケースシナリオの構成図を図 7 へ示す。

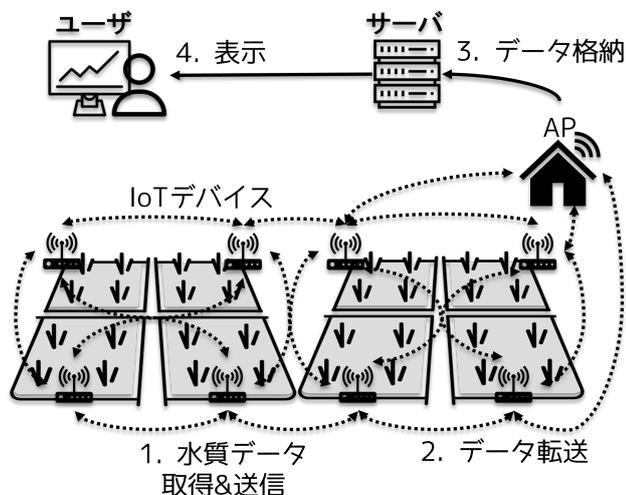


図 7 水田での水質監視ユースケース

図 7 は主に 4 つのコンポーネントと、4 つの手順によって構成されている。図 7 におけるコンポーネントは、水質データを取得と送信するための IoT デバイス、IoT デバイスから受信したデータをサーバへ転送するためのアクセスポイント (AP)、水質データを保存するためのサーバ、そして水質データを確認するユーザがある。IoT デバイスは合計 8 台設置してあるが、この中でアクセスポイントに直接接続できる IoT デバイスは 3 台のみである。そのため、アクセスポイントに直接接続できない IoT デバイスは本提案手法のメッシュネットワークを用いてデータをサーバまで送信する。また IoT デバイスは 1 町\*1の水田あたりにつき 1 台の設置とする。

図 7 では水質データを取得してからユーザへデータを表示するまで、合計 4 つの手順を踏む。これは図 7 中にも示している。まず水田に設置されている IoT デバイスにて水質データである水温と水位データを取得する。そして得られた水質データは別の IoT デバイスへと送信される。送信された IoT デバイスはサーバまで水質データを送るた

\*1 町とは水田の広さを表す単位で 1 つである。面積にして 9917.4m<sup>2</sup>となる。

めに、そのデータを転送する。いくつかの IoT デバイスを経由し、AP へ転送する事でサーバへとデータが送信される。サーバでは水質データを受信すると、サーバ内のデータベースへとデータが格納される。そして最終的にユーザ (水田管理者) へとデータが表示される。

本ユースケースにおいて、本稿の提案を用いる事でユーザある水田管理者はより容易に水田の水管理を行うことができる。水位管理は稲作の肝となる。水位の確認を行うために、水田管理者 (農家) は 30 分から 1 時間間隔でその水田まで向かい直接視認するしかない。しかしこの作業はこまめに自宅と水田を往復する必要がある、肉体的にも時間的にも負担となる。この課題を解決するために本稿の提案手法は IoT デバイスを電力供給インフラやネットワークインフラをコストをかけて整える必要なく、展開することができ、低コストで水田水質監視を行うことが可能となる。

## 3. 実装と実験方法

本章では提案を実装した環境や各種設定やソフトウェア構成図について述べる。

### 実装

本節では提案の実装手法について述べる。実装はサーバと IoT デバイスである ESP-WROOM-32 の 2 つのハードウェアを用いて行う。またサーバと ESP-WROOM-32 はネットワークにて接続されている。図 8 ではサーバと ESP-WROOM-32 の接続における実装構成図を示している。

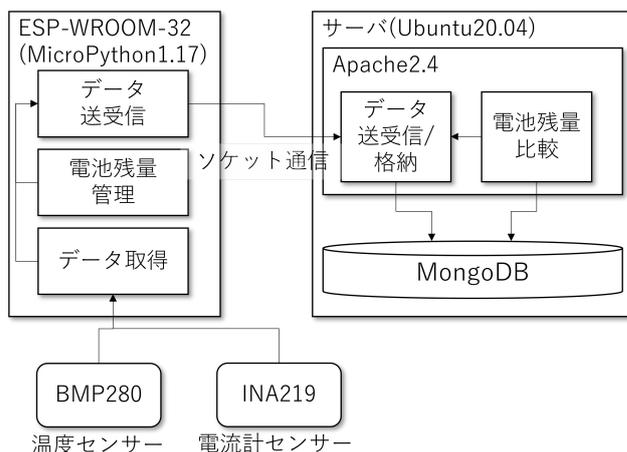


図 8 ESP-WROOM-32 とサーバの接続における実装構成図

図 8 における ESP-WROOM-32 のデータ送受信は ESP-WROOM-32 からサーバの通信に必要な処理を行っている。また ESP-WROOM-32 とサーバとの通信ではソケット通信を用いる。また本稿における実装では ESP32 に電池を取り付けず、実際の電池残量の計測を実施しない。そのため電池残量を予め定め、INA219 にて計測した消費電

流を電池残量から減算していき、電池残量の推移を求める。この処理を行うのが電池残量管理である。またデータ取得は温度センサーである BMP280 と電流計センサーである INA219 からそれぞれデータを取得するための処理を行う。

また図 8 におけるサーバは Web サーバとして Apache2.4 がインストールされている。またデータベースには MongoDB を用いる。データ送受信/格納は ESP-WROOM-32 から送信されたデータをデータベースへ格納する。もしくは電池残量を比較し ESP-WROOM-32 へ中継機能を起動または停止要求データを送信する処理を行う。電池残量比較は ESP-WROOM-32 から送信される電池残量のデータの比較を行う。

### 実験環境

表 1 は実装におけるハードウェアとソフトウェアを示している。

IoT デバイス	ESP-WROOM-32
ファームウェア	MicroPython1.17
電力計センサー	INA219
温度センサー	BMP280
サーバ	Ubuntu20.04
Web サーバ	Apache2.4
データベース	MongoDB 4.2.11

表 1 実装におけるハードウェアとソフトウェアの詳細

図 9 に実験構成図を示す。実験では ESP-WROOM-32 を 3 台とし、それぞれ ESP32-1, ESP32-2, ESP32-3 とする。またそれぞれの ESP-WROOM-32 からは温度データを Wi-Fi ルータを経由してサーバへ送信する。

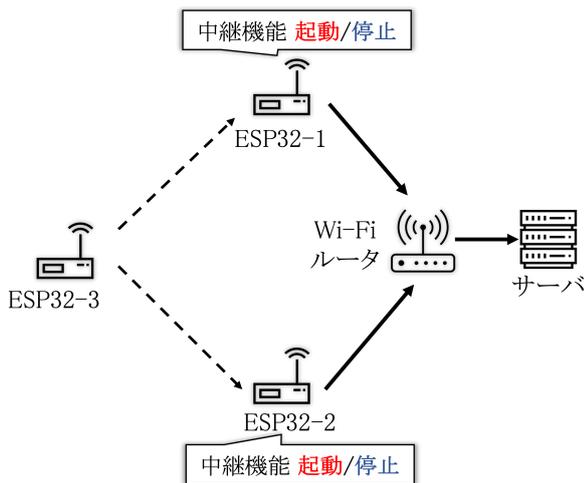


図 9 実験構成図

ESP32-1 と ESP32-2 は ESP32-3 のセンサーデータの中継する必要がある。実験では ESP32-1 と ESP32-2 の電池残量をサーバにて比較し、どちらか一方の ESP-WROOM-32 の中継機能を起動し、もう一方の ESP-WROOM-32 の

中継機能を停止する。

## 4. 評価と分析

評価では提案手法と既存手法を比較し、ESP-WROOM-32 の動作時間と電池残量の推移を比較する。本稿における既存手法はデータを転送する必要がある状態であっても中継機能を起動させ続けている状態のことを指している。図 10 に評価結果を示す。提案手法により既存手法と比べて ESP32-1 の動作時間は 165 秒増加し、ESP32-2 は 180 秒増加した。また消費電力は ESP32-1 にて 13.5mAh 減少し、ESP32-2 では 16.2mAh 減少した。

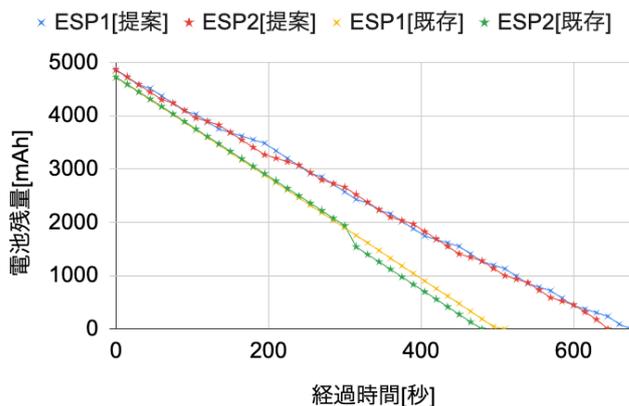


図 10 評価グラフ

## 5. 議論

本稿における手法は中継機能の起動と停止を繰り返し行うが、そのタイミングが電池残量の順位が変わった際に設定されているため、切り替えコストにより更に消費電力が増加してしまう。そのため切り替えるタイミングをあらかじめ電池残量の減少推移から見積もりを行い動的に行う必要がある。

## 6. おわりに

本稿では電池で動作する IoT デバイスにてメッシュネットワークを構築する際に、センサーデータを転送する必要のない場合に中継機能を起動していることによって削減することができる消費電力を削減できないことを課題とした。この課題を解決するため中継機能を電池残量に基づいて起動と停止を行い、IoT デバイスの省電力化の手法を提案した。これにより既存手法と比較し動作時間は最大 180 秒伸び、消費電力は最大 16.2mAh 減少することが実現できた。

## 参考文献

- [1] Sreeram, M. and Sreeja, M.: A novel architecture for IoT and smart community, *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and*

*Cloud* (*I-SMAC*), pp. 486–491 (online), DOI: 10.1109/I-SMAC.2017.8058398 (2017).